

## Release Features & Bugs Quality Gate Guidelines

### Criteria for Features:

1. **Ask:** how risky is it to merge the feature or improvement?

| COMPLETE BY THIS DATE      | DETAILS   |
|----------------------------|---|
| T-15 - Submitting features | New features must be submitted by this date (especially major ones).<br>-e.g. Advanced Permissions<br>-e.g. DND indicator   |
| T-12 - Feature complete    | - Both new features and major improvements must be merged by this date.<br><br>- Minor improvements can be <i>submitted</i> anytime, as long as they are merged by this date (e.g. change colour of DND indicator). |
| T-9 - Code complete        | - Any improvements that don't change the product in any way. (This is rare)   |
| T-7 - Testing done         | N/A   |
| T-2 - Release cut          | N/A   |

### Criteria for Bugs:

1. % of customers affected
2. **“scale”** of impact
  - a. S1 - Prevents app use (e.g. Mattermost crash when posting a message) OR functional regressions
  - b. S2 - 50%-100% of users' functionality affected (e.g. Cannot favorite a channel) OR cosmetic/UI regressions
  - c. S3 - 20%-50% of users' functionality affected (e.g. Cannot create custom emoji if System Admin)
  - d. S4 - Other minor (e.g. channel ID is too small in View Info modal)

| COMPLETE BY THIS DATE       | DETAILS  |
|-----------------------------|--|
| Feature complete            | Focus on features, but bugs that prevent app use (S1 - e.g. Mattermost crashing) good to fix ASAP. |
| Code complete               | S1, S2, S3 and S4 bugs should be fixed by this date.   |
| RC-1 cut and testing begins | S1, S2 and S3 bugs fixed by this date.   |
| Testing done                | S1 and S2 bugs that are found during testing fixed by this date.                                   |
| Release cut                 | S1 bugs - those that are found later will go to dot release.                                       |

- Should bugs be categorized based on:
  - Whether they are P1, P2, P3.
  - How risky they are to fix.
  - Whether it's easier to revert code.
  - Whether it's a regression.
  - How many users it impacts.
- How do we find the time to go through each bug in detail during triage meetings?
  -
- At what point of the release cycle should we stop fixing bugs?
  -

1. After code complete, only fix regressions in the current release.
2. Add "Code Freeze" day after which only P1 regressions are fixed: crashes, security vulnerabilities, etc. This would ideally be T-5.
3. Add "Severity Level" field in Jira
  - a. Document the levels of severity (S1, S2, S3)

#### Criteria for Bugs elsewhere (docs, marketing):

1. % of customers affected
2. "scale" of impact

- e. S1 - Functional regressions (e.g. link not working, breaking change note missing, config.json setting missing).
- f. S2 - Cosmetic/UI regressions (e.g. image blurry)
- g. S3 - Minor issues

t of my propos